Java语言技术管理规范文档

# 一、命名规范

## a. 强制

* 类名使用大驼峰命名法（PascalCase），如：StudentManager。
* 变量名、方法名使用小驼峰命名法（camelCase），如：getStudentList。
* 常量名必须全部大写，并用下划线分隔，如：MAX\_VALUE。
* 包名必须使用小写字母，不得使用下划线，例如：com.company.project.module。
* 禁止使用拼音、无意义的缩写命名，如变量名 aaa, temp1, shuzi 等。

## b. 推荐

* 类名应使用名词，方法名应使用动词或动宾结构。
* 布尔类型变量应以 is、has、can 等前缀开头，如：isDeleted。
* 接口命名应使用 I 前缀或以 able 结尾，如：IUserService 或 Serializable。
* 枚举类型使用复数形式，表示多个状态，如：UserStatus。
* 私有变量可使用 m 前缀或加 \_（下划线），例如：mUserName 或 \_userName。

## c. 允许

* 特殊情况下可以使用简写，但必须是广泛认知的缩写，如：URL, HTML, ID。

# 二、代码结构与格式

## a. 强制

* 每个类文件只能包含一个顶级 public 类，类名需与文件名一致。
* 类、方法之间必须空一行。
* 使用 4个空格缩进，禁止使用Tab。
* 所有 if/else/for/while 语句必须加大括号 {}。
* switch 语句必须包含 default 分支。

## b. 推荐

* 方法行数建议不超过 50 行，超过应拆分。
* 每个类尽量只负责一个功能（单一职责原则）。
* 每个源文件推荐在开头写明作者、创建时间、用途等注释信息。
* 推荐使用 try-with-resources 进行资源管理（如关闭流）。
* 尽量不要在一个类中定义多个嵌套类。

## c. 允许

* 对于非常简单的 get/set 方法可不写注释。

# 三、注释规范

## a. 强制

* 所有 public 方法必须添加 JavaDoc 注释，说明方法的功能、参数和返回值。
* 注释不得出现与代码逻辑不一致的内容，必须及时更新。
* 禁止无意义的注释，如 // TODO、// 变量a 等空注释。

## b. 推荐

* 推荐使用多行注释 /\* \*/ 来描述复杂逻辑或模块说明。
* 方法内的关键逻辑段落建议使用行注释 // 说明。
* 推荐使用 @deprecated 注明不推荐使用的方法。

## c. 允许

* 可以使用中文注释，但应避免中英文混杂。

# 四、异常与日志

## a. 强制

* 禁止捕获 Exception 或 Throwable，必须捕获具体异常类型。
* 捕获异常后必须处理，不得空 catch。
* 不允许在代码中使用 System.out.println 打印调试信息，必须使用日志框架。

## b. 推荐

* 日志级别应分清：DEBUG、INFO、WARN、ERROR。
* 抛出异常时应附带描述信息，如：throw new IllegalArgumentException("参数错误")。
* 日志信息应可定位问题，包含关键参数、上下文信息。

## c. 允许

* 在测试环境中可以适度使用标准输出，部署前应删除。

# 五、性能与安全

## a. 强制

* 禁止在循环中创建对象或连接数据库。
* 禁止拼接 SQL 字符串，必须使用预编译参数。
* 用户输入必须做参数校验，防止注入和越权。

## b. 推荐

* 尽量使用线程池，而不是手动创建线程（如 new Thread()）。